VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

STAGGERED STRONG COUPLING BETWEEN EXISTING FLUID AND SOLID SOLVERS THROUGH A PYTHON INTERFACE FOR FLUID-STRUCTURE INTERACTION PROBLEMS

David Thomas*, Anil Variyar†, Romain Boman*, Thomas D. Economon†, Juan J. Alonso†, Grigorios Dimitriadis* and Vincent E. Terrapon*

*Department of Aerospace and Mechanical Engineering
University of Liège
Liège, 4000, Belgium
e-mail: {dthomas, r.boman, gdimitriadis, vincent.terrapon}@ulg.ac.be

†Department of Aeronautics and Astronautics
Stanford University
Stanford, CA, 94305, USA
e-mail: {anilvar, economon, jjalonso}@stanford.edu

Key words: Fluid-Structure Interaction, Strong Coupling, Python Wrapper, Computational Aeroelasticity.

Abstract. A Fluid-Structure Interaction (FSI) tool that couples existing independent fluid and solid solvers into a single synchronization and communication framework based on the Python language is presented. Each solver has to be wrapped in a Python layer in order to embed their functionalities (usually written in a compiled language) into a Python object, that is called and used by the coupler. Thus a staggered strong coupling can be achieved for time-dependent FSI problems such as aeroelastic flutter or vortex-induced vibrations (VIV). The synchronization between the solvers is performed with the block-Gauss-Seidel algorithm and a dynamic under-relaxation. The tool allows non-matching meshes between the fluid and structure domains and it is optimised to work in parallel using Message Passive Interface (MPI). These capabilities are demonstrated on typical validation cases. The open-source code SU2 is used to compute the fluid region while the solid region is computed either by a simple rigid body integrator, by an in-house nonlinear Finite Element code (Metafor) or by the structural solver TACS. First, the accuracy of the results is demonstrated and then the modularity of the coupling as well as its ease of use is highlighted.

1 INTRODUCTION

The computation of Fluid-Structure Interaction (FSI) problems is usually based on one of two possible strategies: the monolithic or the partitioned approach [1, 2, 3]. In the monolithic approach, both the structural and fluid problems are solved within one single solver and within the same mathematical framework where the interfacial conditions are implicit to the procedure. On the other hand, the partitioned approach couples two different specialized existing codes. This requires an efficient communication and synchronization framework, but allows the intrinsic features of the individual solvers to be leveraged.

A framework to couple two heterogeneous existing fluid and solid solvers is presented here. It is based on a Python coupling environment designed to interact with the two solvers through their respective Python wrappers. Data exchange and synchronization are thus implemented in a very intuitive and flexible way in Python, whereas the computationally intensive routines within each solver are kept in their native languages (C, C++, ...). Because the coupled solvers are reduced to black-box tools by their wrappers, minimal effort is required to ensure compatibility with the coupler, providing great flexibility regarding the solvers that can be coupled.

The paper is organized as follows. Section 2 is dedicated to describing the governing equations, the computational implementation of the fluid/structural mechanics and the coupling conditions. The implementation of the coupling environment is then presented in Section 3. In Section 4 several FSI test cases are reproduced and results are presented in order to show the accuracy and the flexibility of the coupling tool. Finally, Section 5 summarizes the main concepts and results, and suggests further steps for future work.

2 GOVERNING EQUATIONS OF THE COUPLED PROBLEM

This section summarizes the governing equations and their numerical implementation for both the fluid and solid parts of the coupled problem. The fluid equations in an Arbitrary Lagrangian-Eulerian (ALE) form are solved in a moving domain Ω_f that shares a common boundary Γ with the solid domain Ω_s, in which the solid equations are solved with a Lagrangian formalism. In addition to intrinsic boundary conditions for each of the disciplines, coupling conditions on the displacements, and the loads across the common boundary Γ, are required to achieve a strong coupling scheme.

2.1 Fluid mechanics

The dynamic behavior of a compressible Newtonian fluid is predicted by solving the Navier-Stokes or Euler equations. The conservation of mass, momentum and energy in Ω_f can be written using ALE formalism as [4]

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}^\mathrm{c} - \nabla \cdot (\mathbf{V}_\Omega \mathbf{U}) - \nabla \cdot \mathbf{F}^\mathrm{v} = \mathbf{Q} \qquad \text{in } \Omega_\mathrm{f} \times [0, T] \,, \tag{1}$$

where the conservative variables are given by $\mathbf{U} = [\rho, \rho\mathbf{v}, \rho E]^{\mathrm{T}}$, $\mathbf{F}^{\mathrm{c}}$, $\mathbf{V}_{\Omega}\mathbf{U}$ and $\mathbf{F}^{\mathrm{v}}$ are the advective, ALE and diffusive fluxes, respectively, and $\mathbf{Q}$ is a source term. In these expressions ρ is the fluid density, $\mathbf{v} = [v_1, v_2, v_3]^{\mathrm{T}}$ the velocity field in Cartesian coordinates, E the total energy per unit mass and $\mathbf{V}_{\Omega} = [\mathbf{v}_{\Omega}, \mathbf{v}_{\Omega}, \mathbf{v}_{\Omega}]^{\mathrm{T}}$ the local velocity of the moving domain. The Euler equations are simply recovered by discarding the viscous terms. Note that in the case of a turbulent viscous flow, the Reynolds-Averaged Navier-Stokes (RANS) equations with a turbulence model (e.g., k-ω SST [5], Spalart-Allmaras [6]) could also be solved.

The open-source CFD code SU2 [7, 8] is used to solve the fluid part of the coupled problem. The governing equations are spatially discretized using the Finite Volume Method on a dual-grid using a vertex-based approach. Temporal discretization is achieved through a dual time-stepping [9] strategy, where each physical time step is transformed into a steady problem.

2.2 Solid mechanics

The dynamic behavior of a deformable solid results from the balance between inertial, internal and external forces. The equilibrium equation in Ω_{s} is given by

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} - \nabla\cdot\overline{\overline{\sigma}} = \mathbf{f} \qquad \text{in } \Omega_{\mathrm{s}}\times[0,T]\ , \tag{2}$$

where ρ, $\mathbf{u}$, $\overline{\overline{\sigma}}$ and $\mathbf{f}$ are the solid density, the displacement vector, the Cauchy stress tensor and the body forces, respectively. The in-house nonlinear Finite Element code Metafor [10] is used to solve the structural part of the coupled problem. The solver is designed to simulate large structural deformations by expressing the principle of virtual work (PVW) on the deformed configuration. A particular feature of Metafor is also the large range of nonlinear material laws that can be used (elasticity, elasto-plasticity or visco-elasto-plasticity, ...). For each time increment, these nonlinearities are solved with a Newton-Raphson approach. Time discretization typically uses the Generalized-α method but a quasi-static integration is also available when inertia terms are negligible.

Simpler coupled fluid-structure problems involve the motion of non-deformable solids where the dynamics is constrained using stiffness or damping. Such a model is implemented in a in-house rigid body integrator code also based on the Generalized-α method for time integration.

Also, the coupling approach and the Python wrapper/interface developed for the fluid solver SU2 is further tested by coupling it with the Toolkit for Analysis of Composite Structures (TACS) , which is a linear/nonlinear solution capable structural solver developed by Kennedy and Martins [11]. TACS already has a Python interface which makes coupling it with SU2 easy.

2.3 Coupling conditions

The fluid and solid domains are coupled at their common interface Γ through continuity boundary conditions on the displacement $\mathbf{d}^\Gamma$ and the load $\mathbf{t}^\Gamma$,

$$\begin{aligned} \mathbf{d}_\mathrm{f}^\Gamma &= \mathbf{d}_\mathrm{s}^\Gamma \\ \mathbf{t}_\mathrm{f}^\Gamma &= -\mathbf{t}_\mathrm{s}^\Gamma \,, \end{aligned} \tag{3}$$

where the load on the fluid side is given by $\mathbf{t}_\mathrm{f} = -p\mathbf{n}_\mathrm{f} + \overline{\overline{\tau}}\mathbf{n}_\mathrm{f}$, with p the pressure and $\overline{\overline{\tau}}$ the viscous stress, and the load on the solid side by $\mathbf{t}_\mathrm{f} = \overline{\overline{\sigma}}\mathbf{n}_\mathrm{s}$. The normal unit vectors $\mathbf{n}_\mathrm{f}$ and $\mathbf{n}_\mathrm{s}$ are both pointing outwards from their respective domains.

The coupled problem can be solved using a Dirichlet-Neumann approach [12]. Introducing a Dirichlet nonlinear operator $\mathcal{F}$ that computes the fluid load from a given fluid interface displacement,

$$\mathbf{t}_\mathrm{f}^\Gamma = \mathcal{F}(\mathbf{d}_\mathrm{f}^\Gamma) \,, \tag{4}$$

and a Neumann nonlinear operator $\mathcal{S}$ that computes the solid interface displacement as a function of the solid load,

$$\mathbf{d}_\mathrm{s}^\Gamma = \mathcal{S}(\mathbf{t}_\mathrm{s}^\Gamma) \,, \tag{5}$$

Equation (3) can be formulated as a fixed-point problem [13]:

$$\mathbf{d}^\Gamma = \mathcal{S}\left(-\mathcal{F}(\mathbf{d}^\Gamma)\right) \tag{6}$$

where $\mathbf{d}^\Gamma$ is the displacement common to both the solid and fluid interfaces.

Finally, the mesh within the fluid domain must be adapted to accommodate the solid deformation/displacement. The mesh morphing used with the ALE formalism can be expressed as

$$(\mathbf{x}_\mathrm{f}^\Omega, \mathbf{v}_\mathrm{f}^\Omega) = \mathcal{M}(\mathbf{d}_\mathrm{f}^\Gamma) \,, \tag{7}$$

where $\mathbf{x}_\mathrm{f}^\Omega$ and $\mathbf{v}_\mathrm{f}^\Omega$ are the position and velocity of the mesh points, respectively, and $\mathcal{M}$ a mesh deformation operator.

3 IMPLEMENTATION OF THE COUPLING ENVIRONMENT

The coupling environment for solving the FSI problem, represented by Equations (1), (2) and (6), is implemented using a partitioned framework. The coupler is based on a block-Gauss-Seidel (BGS) algorithm, that synchronizes the solvers [13, 14] in a strong coupling scheme. This ensures that the coupling conditions on the FSI-interface at each time step are met by iterating between the fluid ($\mathcal{F}$) and solid ($\mathcal{S}$) computations. These iterations are repeated until a convergence criterion, based on the norm of the difference in the structural displacement between two successive BGS iterations, is met. $\mathcal{F}$ and $\mathcal{S}$ are reduced to abstract black-box calls to the respective solvers. This is achieved through a modular and high-level implementation of the coupling environment using the Python programming language. Modules of the coupled solvers written in C/C++, are wrapped in

a Python layer that behaves as a driving and communicating channel between the coupler and the solver itself. The wrapping is easily performed using the Simplified Wrapper and Interface Generator (SWIG) tool [15]. For data exchange between the solvers and to call required modules, the main coupler (written in pure Python) can directly interact with the solvers through their wrappers as if they were simple Python objects. These communications do not involve any file i/o. The wrapped functionalities of each solver can thus be easily and intuitively managed in Python while the critical and computationally intensive calculations are performed by the native solvers. This wrapping process and the interaction between the solvers and the coupling environment are schematically illustrated in Figure 1.

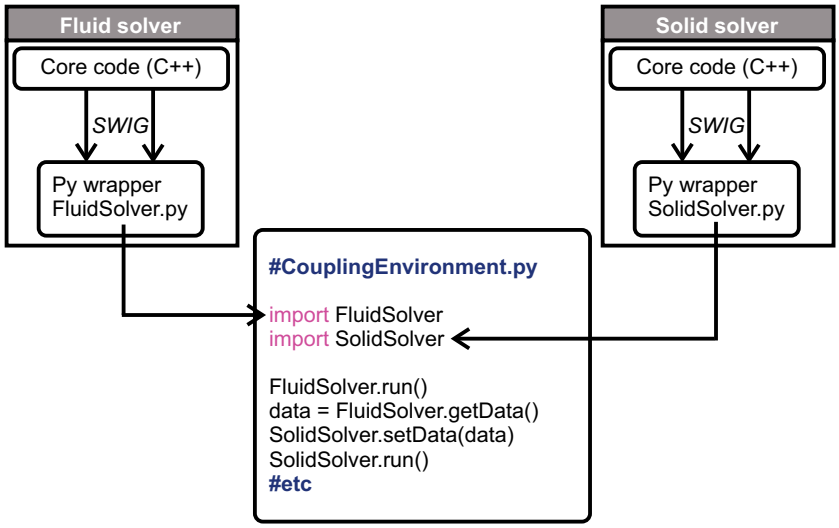

Figure 1: Schematic illustration of the coupling environment and its interaction with the respective fluid and solid Python wrappers.

This generic high-level coupling environment ensures both modularity and flexibility (see Section 4 for examples). In particular, there is no specific restriction on the solvers (fluid or solid) that can be used, except for minimal compatibility requirements between their wrappers and the coupler (e.g. exchanged data must be expressed in a typical array format).

The coupler can also handle parallelized fluid/solid solvers in a very flexible way using the MPI paradigm and its Python bindings mpi4py [16]. The communications between different partitions are performed by the coupler itself (except for intrinsic MPI communications within each solver) before communicating with the solvers. Moreover, the MPI processes can be distributed without any restriction, i.e., one process can instantiate either a fluid or solid partition or even both (since the staggered coupling requires a sequential run of the solvers). Mixing parallel and serial solvers is thus possible. Finally, note that bindings for PETSc (petsc4py) [17] can be interfaced for all parallel linear

algebra operations.

The BGS approach used to couple the solvers is directly implemented in the Python coupler. In order to handle ill-conditioned cases such as those involving strong added-mass effects, dynamic under-relaxation using Aitken's formulation is implemented [13]. A second order predictor [18] is used between time steps to accelerate the convergence of the BGS loop.

Interface interpolations are performed at the coupler level. In a partitioned approach solid and fluid meshes are likely to be created independently of each other for optimality. Consequently, there is no guarantee that the boundary discretization at the fluid-structure interface consist of matching meshes. The data transferred between the two solvers must therefore be interpolated from one grid to the other. This is equivalent to defining a new operator $\mathcal{I}_{\mathrm{s}}^{\mathrm{f}}$ that maps the displacements of the solid interface mesh onto the fluid interface mesh during the communication step of the BGS algorithm,

$$\mathbf{d}_{\mathrm{f}}^{\Gamma} = \mathcal{I}_{\mathrm{s}}^{\mathrm{f}}(\mathbf{d}_{\mathrm{s}}^{\Gamma})\,, \tag{8}$$

or, analogously, an operator $\mathcal{I}_{\mathrm{f}}^{\mathrm{s}}$ that maps the load from the fluid to the solid,

$$\mathbf{t}_{\mathrm{s}}^{\Gamma} = \mathcal{I}_{\mathrm{f}}^{\mathrm{s}}(\mathbf{t}_{\mathrm{f}}^{\Gamma})\,. \tag{9}$$

These operators can be expressed as simple linear algebraic interpolation matrices [19]. Currently the Radial Basis Function (RBF) interpolation technique [20], using either a local $\mathcal{C}^2$ or a global Thin Plate Spline (TPS) function, is implemented in the coupler. The overall coupling algorithm is illustrated in Figure 2.

4 RESULTS

The coupler implementation described in the previous section is now used to solve several FSI test cases in order to demonstrate its accuracy, flexibility and robustness.

4.1 Isogai wing section

The coupling between SU2 and the rigid body integrator is tested using the classical Isogai wing section aeroelastic case (case A) [21, 22]. This test case represents the dynamics of the outboard portion of a swept-back wing in the transonic regime. The airfoil is a symmetric NACA 65a010 profile with a chord $c = 2b$. As shown in Figure 3, the displacement h of the elastic axis is positive downwards and the pitch angle α is positive clockwise. The static unbalance S is defined as the product of the airfoil mass m with the distance $x_{\mathrm{CG}} - x_{\mathrm{f}}$ between the center of gravity and the elastic axis. The structural dynamics is modeled with a spring-dashpot system with stiffnesses K_h and K_α and damping coefficients C_h and C_α for the plunging and pitching mode.

The equations of motion for this aeroelastic system can be written as [23]

$$\begin{aligned} m\ddot{h} + S\ddot{\alpha} + C_h\dot{h} + K_h h &= -L\,, \\ S\ddot{h} + I_{\mathrm{f}}\ddot{\alpha} + C_\alpha\dot{\alpha} + K_\alpha\alpha &= M\,, \end{aligned} \tag{10}$$

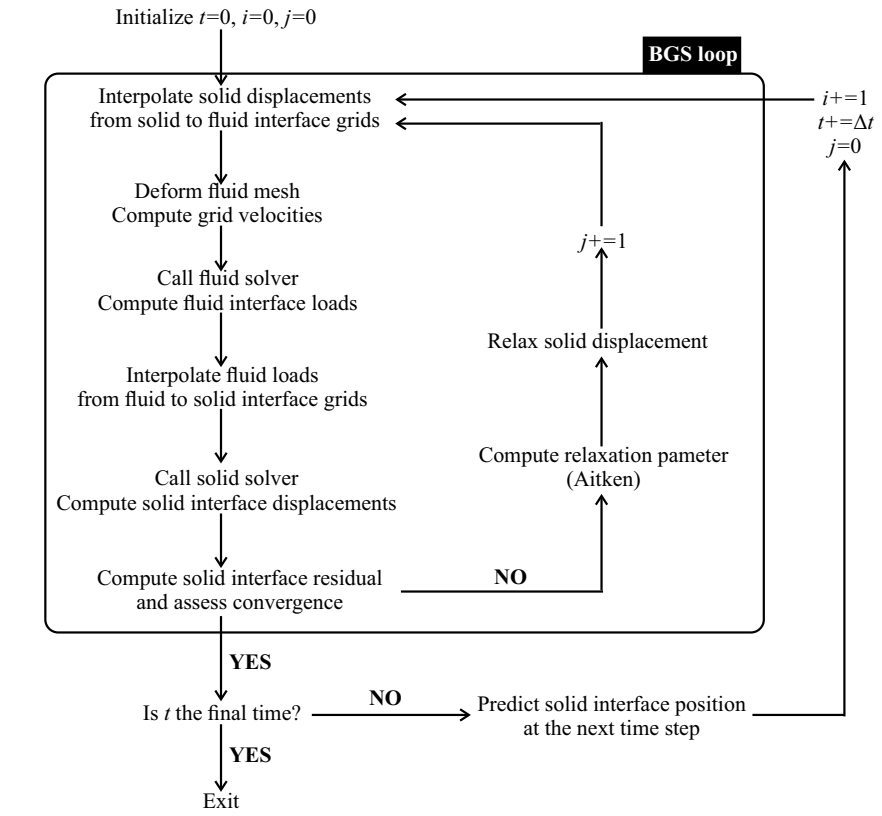

Figure 2: Time-marching coupling algorithm based on a block-Gauss-Seidel scheme (i is the time iterator and j is the FSI iterator).

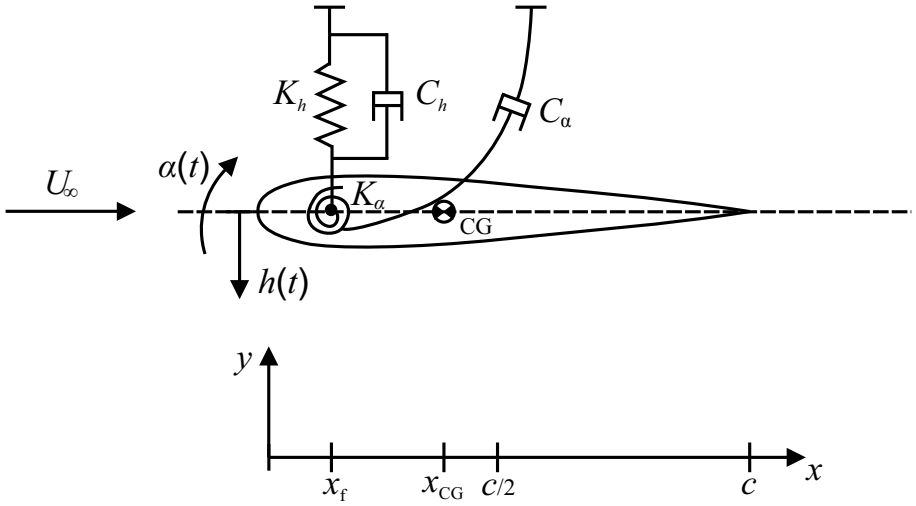

Figure 3: Schematic of a two degrees of freedom pitching-plunging airfoil aeroelastic model.

where I_f is the moment of inertia of the airfoil around the elastic axis, L the aerodynamic lift (positive upwards) and M the aerodynamic moment with respect to the elastic axis (positive clockwise). The overall system is characterized by several non-dimensional parameters, i.e., the normalized static unbalance $\chi = S/mb$ and moment of inertia $r_\alpha^2 = I_\mathrm{f}/mb^2$, the plunging and pitching damping ratios $\eta_h = C_h/2\sqrt{K_h m}$ and $\eta_\alpha = C_\alpha/2\sqrt{K_\alpha I_\mathrm{f}}$, the mass ratio $\mu = m/\pi\rho_\infty b^2$ where ρ_∞ is the free-stream fluid density, and the natural frequency ratio $\overline{\omega} = \omega_h/\omega_\alpha$ where $\omega_h = \sqrt{K_h/m}$ and $\omega_\alpha = \sqrt{K_\alpha/I_\mathrm{f}}$ are the natural frequencies of the uncoupled system. The parameters for the Isogai test case are $\chi = 1.8$, $r_\alpha = 1.865$, $\overline{\omega} = 1$ and $\mu = 60$. There is no structural damping, i.e., $C_h = C_\alpha = 0$. The elastic axis is placed in front of the airfoil at a distance $x_\mathrm{f} = -b$ from the leading edge and the natural pitching frequency is here $\omega_\alpha = 100$ rad/s. The Euler equations are solved in the transonic regime with an initial airfoil pitch angle $\alpha_0 = 0.0174$ rad (1^o). The time step for the simulation is $\Delta t = 0.0016$ s which corresponds to 39 time steps per period of the uncoupled pitch mode. Because of the high mass ratio of the coupled system, low added mass effects are expected, thus no relaxation is used. Three BGS iterations per time step are typically required to achieve a coupling tolerance of 10^{-6} m (10^{-4} times the initial perturbation) on the structural displacement.

Several FSI simulations at different transonic free-stream Mach numbers ($\mathrm{M}_\infty = 0.7 - 0.9$) are performed with variable speed index

$$V^* = \frac{U_\infty}{b\omega_\alpha\sqrt{\mu}} \tag{11}$$

in order to predict the flutter point. Flutter inception is identified as the point for which the damping extracted from the system dynamic response is zero. The computed flutter speed indices are compared with values from the literature [24, 25, 26, 27], as shown in Figure 4. The best approximation curve (spline) is a representation of the limit between the stable and unstable regions. It can be seen that the "transonic dip" and the typical "S-shape" flutter boundary for M_∞ between 0.7 and 0.9 are both well predicted.

4.2 VIV of a flexible cantilever in the wake of a square cylinder

The study of the flexible cantilever attached to the downstream side of a perfectly rigid square cylinder is a classical two-dimensional benchmark test case for FSI [18]. The geometry of the computational domain is described in Figure 5. In this case $H = 0.01$ m. The physical properties of the solid and fluid are summarized in Table 1. The incoming flow velocity is $U_x = 0.513$ m/s, which corresponds to a laminar Reynolds number $\mathrm{Re} = U_x H/\nu_\mathrm{f} = 333$. The top and bottom sides of the channel are modeled as inviscid walls.

The velocity and the Reynolds number are such that an unsteady laminar Von Karman vortex sheet is generated in the wake of the cylinder with a shedding frequency close to the first bending frequency of the flexible cantilever. Therefore, the vortical structure of the wake generates harmonic aerodynamic loads that induce periodic oscillations of the flexible cantilever.

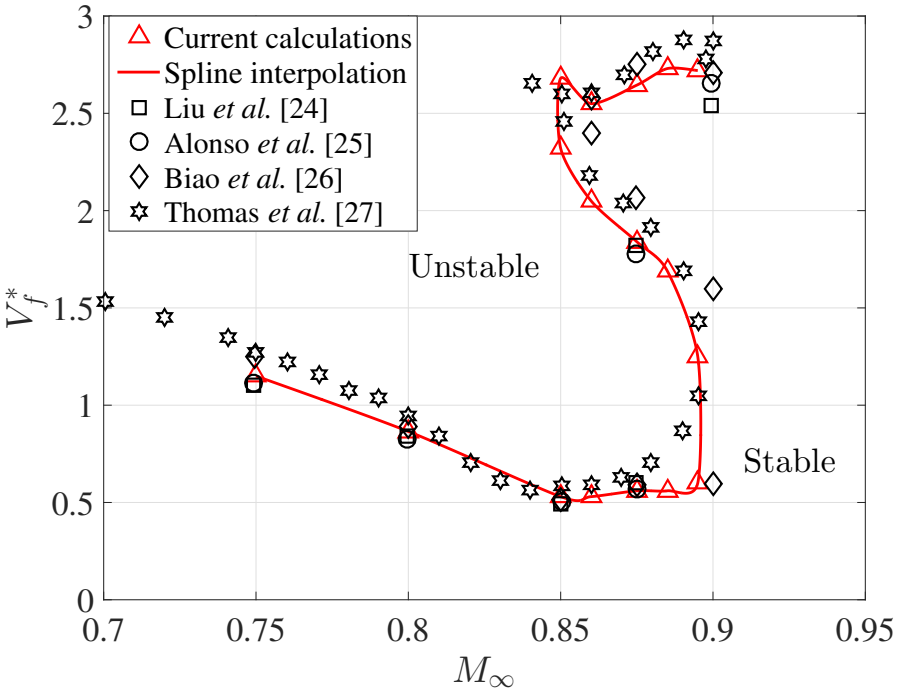

Figure 4: Flutter speed index as a function of the free-stream Mach number. Comparison between current computations and numerical results from the literature.

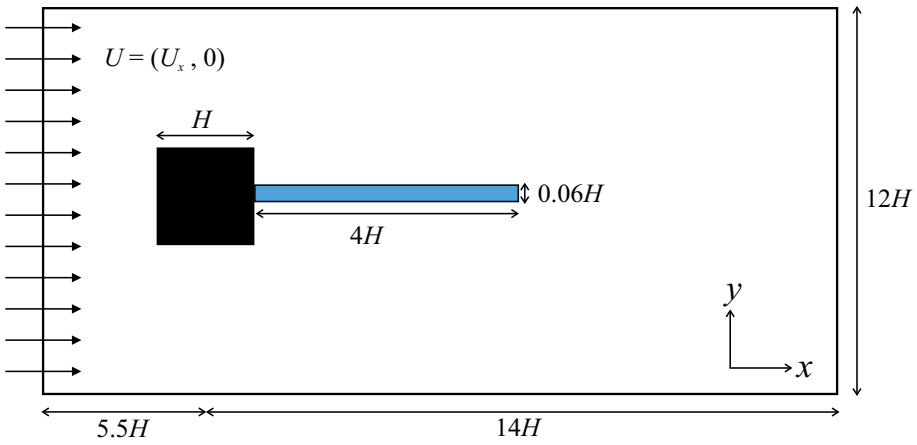

Figure 5: Flexible cantilever attached to a rigid square cylinder. Geometry of the computational domain.

This interaction is numerically reproduced by coupling SU2 and Metafor. A nonlinear formalism is necessary to correctly predict the bending of the cantilever undergoing large displacements. In order to eliminate any interpolation error, the grid discretization is done

Solid		
Density [kg m^{-3}]	ρ_s	100
Young's modulus [Pa]	E	$2.5 \cdot 10^5$
Poisson's ratio [-]	ν_s	0.35
Fluid		
Density [kg m^{-3}]	ρ_f	1.18
Kinematic viscosity [m^2 s^{-1}]	ν_f	$1.54 \cdot 10^{-5}$

Table 1: Solid and fluid physical properties.

so as to have matching meshes at the fluid-structure interface. The mesh is structured in the near-field of the solid bodies (square cylinder and cantilever) and then unstructured throughout the rest of the domain. The time step for the simulation is $\Delta t = 0.0025$ s which corresponds to 122 time steps per period on the first bending mode of the beam. Four BGS iterations with no relaxation are typically required to reach a coupling tolerance of 10^{-6} m (10^{-4} times the expected tip displacement) on the structural displacement.

Habchi *et al.* [18] summarized results from the literature (e.g., [28, 14, 29]). The oscillations frequency typically falls in the range $2.94 - 3.25$ Hz, while the amplitude of the tip displacement is in the range $0.95 - 1.25$ cm. The present computation predicts a maximum tip displacement $d_y = 1.07$ cm and a frequency $f = 3.14$ Hz , which is in very good agreement with results from the literature. The same simulation is repeated using the SU2-TACS framework and a maximum tip displacement $d_y = 1.03$ cm and an average frequency of $f = 2.93$ Hz is observed, matching well with the SU2-Metafor simulation and available data. Figure 6 shows the velocity magnitude contour at several time steps of a period T, where the vortical structures and the large displacement of the cantilever can be observed.

4.3 Static aeroelastic deformation of the AGARD 445.6 wing

The experimental AGARD 445.6 wing test case [30] is a frequently used three-dimensional validation case for transonic flutter simulations. The present computational study is based on the weakened model 3 of the wing. This is a 45° swept-back wing whose geometrical properties are summarized in Table 2. The cross section of the wing is a symmetric NACA 65a004 airfoil. The static aeroelatic case performed by Goura [31] is here reproduced by solving the Euler equations with SU2. The wing is modeled in Metafor with 8-node continuum elements and an orthotropic elastic material whose properties are summarized in Table 3. A modal analysis is first performed and the first four natural frequencies computed are compared with results in the literature in Table 4, showing good agreements with both computational and experimental results. Note that the present results are obtained with volume finite elements, while the results from the literature were based on plate finite elements, which could explain the small discrepancies observed in

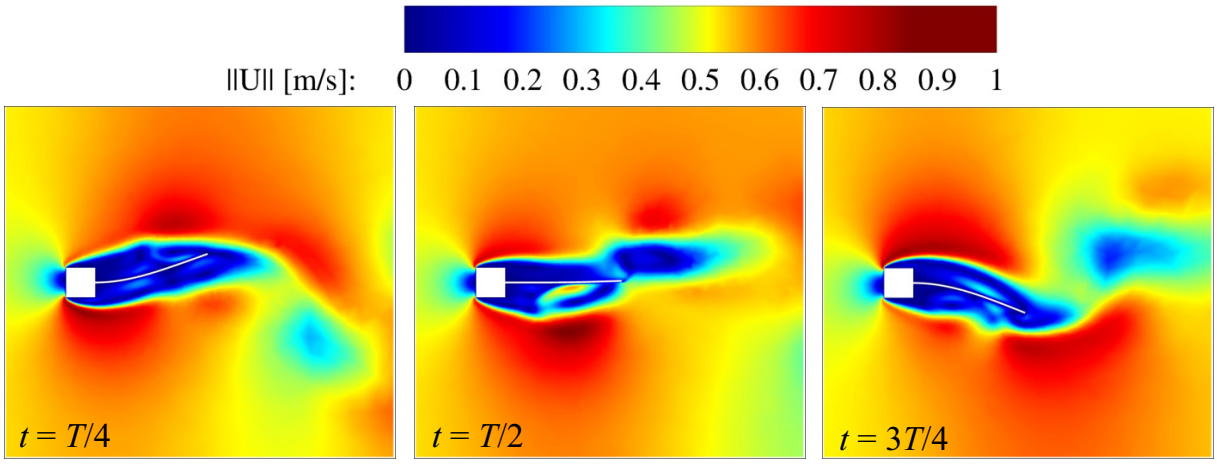

Figure 6: Velocity magnitude contour at three different times of a period. Results from the coupling between SU2 and Metafor.

Root chord [m]	c_r	0.559
Taper ratio [-]	λ	0.658
Tip chord [m]	c_t	0.368
Semi-span [m]	b_s	0.762
Aspect ratio [-]	AR	1.644
Wing surface [m^2]	S	0.353
Mean aerodynamic chord [m]	$\bar{c}$	0.470

Table 2: Geometrical properties of the AGARD 445.6 wing.

Longitudinal Young's modulus [GPa]	E_1	3.151
Transverse Young's moduli [GPa]	E_2, E_3	0.4162
Shear moduli [GPa]	G_{12}, G_{13}, G_{23}	0.4392
Poisson's ratio [-]	$\nu_{12}, \nu_{13}, \nu_{23}$	0.31
Density [kg m^{-3}]	ρ_s	381.98

Table 3: Material properties for the AGARD 445.6 wing.

the natural frequencies.

The coupled simulation is performed in the transonic regime with a free-stream Mach number $M_\infty = 0.8$ at an angle of attack $\alpha = 1°$. The free-stream velocity U_∞ and density ρ_∞ are set to 247.09 m/s and 0.09411 kg/m^3, respectively. A RBF interpolation using the TPS method is used to interpolate data between the solid and fluid meshes at the

	f_1	f_2	f_3	f_4
Present	9.54	40.35	50.22	97.67
Yates* [30]	9.60	38.10	50.70	98.50
Goura [31]	9.67	36.87	50.26	90.00
Beaubien *et al.* [32]	9.46	39.44	49.71	94.39
Zhang*et al.* [33]	9.57	38.17	48.35	91.55

Table 4: First four natural frequencies of the AGARD 445.6 wing from the present calculation and the literature (experimental results are identified by the * symbol). Frequencies are in Hz with f_1 and f_3 corresponding to the first and second bending modes, and f_2 and f_4 to the first and second torsion modes, respectively.

interface. The coupling tolerance is set to 10^{-6} m, which corresponds to 10^{-4} times the expected maximum tip displacement of the wing.

The static aeroelastic deformation at these conditions is illustrated in Figure 7. The values of the vertical displacement of the wing computed at the leading and trailing edges are summarized and compared with other references in Table 5. A discrepancy of 0.4 mm and 0.4-0.9 mm are observed at the leading and trailing edge, respectively, which could be explained by the use here of volume rather than plate finite elements.

	Leading edge	Trailing edge
Present	0.0116	0.0131
Goura [31]	0.0112	0.0127
Melville *et al.* [34]	0.0112	0.0122

Table 5: Vertical displacement of the wing tip at the leading and trailing edges. Values are in m.

5 CONCLUSION

A modular and flexible implementation of a coupling environment for Fluid-Structure Interaction problems was presented. The coupled problem is solved using a partitioned approach in which the fluid and solid solvers communicate through the coupler through a Python layer/interface. This ensures that the high-level management of the two solvers (black-box tools) is very intuitive and flexible and all the intensive calculations remain embedded in their respective core codes.

The results of several test cases were compared to the literature, demonstrating the accuracy of the coupling tool. The high modularity of the framework was demonstrated by using different structural solvers. Moreover, the coupling of other existing solid or

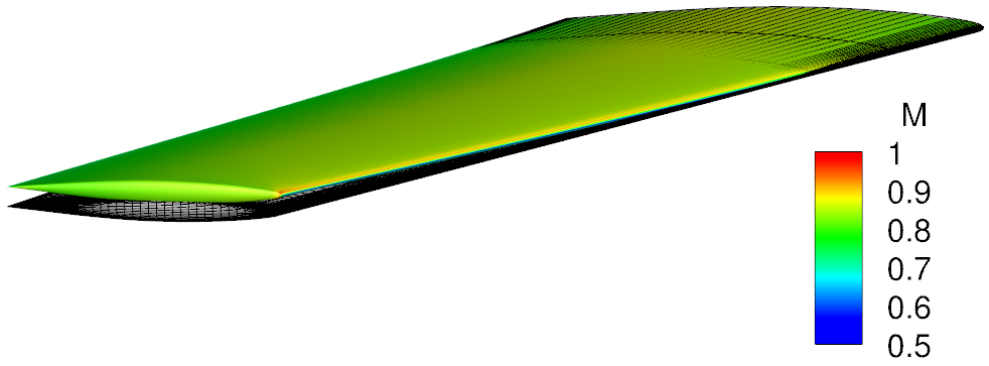

Figure 7: Static aeroelastic deformation of the AGARD 445.6 wing. The initial configuration is in gray. Contours of the Mach number are superposed to the deformed wing.

fluid solvers would be rather straightforward, requiring only minimal adaptations. The framework could also be easily extended to permit the coupling of other physics, such as conjugate heat transfer.

Future work will focus on extending some of the current capabilities of the tool, including the coupling algorithm, and the interpolation methods. Additionally, other multiphysics problems like conjugate heat transfer and coupling with different solvers will also be considered.

REFERENCES

[1] A.Y. Tang and N. Amin. Some numerical approaches to solve fluid structure interaction problems in blood flow. *Abstract and applied analysis*, 2014.

[2] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12(2):337–377, 2012.

[3] L. Garelli. *Fluid-structure interaction using an arbitrary Lagrangian-Eulerian formulation*. PhD thesis, Universidad Nacional Del Litoral, 2011.

[4] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodriguez-Ferran. *Encyclopedia of Computational Mechanics*, chapter Arbitrary Lagrangian-Eulerian Methods. Wiley, 2004.

[5] F.R. Menter. Zonal two equation $k - \omega$ turbulence model for aerodynamic flows. *AIAA Paper 1993-2906*, 1993.

[6] P.R Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *Recherche aerospatiale*, 1:5–21, 1994.

[7] F. Palacios, M.R. Colonno, A.C. Aranake, A. Campos, S.R. Copeland, T.D. Economon, A.K. Lonkar, T.W. Lukaczyk, T.W.R. Taylor, and J.J. Alonso. Stan-

ford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design. In *AIAA 51st Aerospace Sciences Meeting*, Grapevine, TX, 7-10 January, 2013.

[8] R. Sanchez, H.L. Kline, D. Thomas, A. Variyar, M. Righi, T.D. Economon, J.J. Alonso, R. Palacios, G. Dimitriadis, and V. Terrapon. Assessment of the fluid-structure interaction capabilities for aeronautical applications of the open-source solver SU2. In *ECCOMAS Congress, VII European Congress on Computational Methods in Applied Sciences and Engineering*, Crete Island, Greece, June 2016.

[9] A. Jameson and S. Schenectady. An assessment of dual-time stepping, time spectral and artificial compressibility based numerical algorithm for unsteady flow with applications to flapping wings. *AIAA Paper 2009-4273*, 2009.

[10] METAFOR. A nonlinear finite element code. University of Liège, http://metafor.ltas.ulg.ac.be/.

[11] G.J. Kennedy and J.R.R.A Martins. Parallel solution methods for aerostructural analysis and design optimization. *AIAA Paper 2010-9308*, Sept 2010.

[12] J. Degroote, A. Souto-Iglesias, W. Van Paepegem, S. Annerel, P. Bruggeman, and J. Vierendeels. Partitioned simulation of the interaction between an elastic structure and free surface flow. *Computer Methods in Applied Mechanics and Engineering*, 199:2085–2098, 2010.

[13] U. Küttler and W. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[14] C. Wood, A.J. Gil, O. Hassan, and J. Bonet. Partitioned block-Gauss-Seidel coupling for dynamic fluid-structure interaction. *Computers and Structures*, 88:1367–1382, 2010.

[15] D.M. Beazley. SWIG : An easy to use tool for integrating scripting languages with C and C++. In *4th Tcl/Tk Workshop*, Monterey, CA, USA, July 1996.

[16] L. Dalcin, R. Paz, M. Storti, and J. D'Elia. MPI for Python: Performance improvements and MPI-2 extensions. *Journal of Parallel and Distributed Computing*, 68(5):655–662, 2008.

[17] L. Dalcin, R. Paz, P. Kler, and A. Cosimo. Parallel distributed computing using Python. *Advances in Water Resources*, 34(9):1124–1139, 2011.

[18] C. Habchi, S. Russeil, D. Bougeard, J-L. Harion, T. Lemenand, A. Ghanem, D. Della Valle, and H. Peerhossaini. Partitioned solver for strongly coupled fluid-structure interaction. *Computers and Fluids*, 71:306–319, 2013.

[19] A. de Boer, A.H van Zuijlen, and H. Bijl. Review of coupling methods for non-mathcing meshes. *Computational Methods in Applied Mechanics and Engineering*, 196:1515–1525, 2007.

[20] O. Estruch, O. Lehmkuhl, R. Borrel, C.D. Pérez Segarra, and A. Oliva. A parallel radial basis function interpolation for unstructured dynamic meshes. *Computers and Fluids*, 80:44–54, 2013.

[21] K. Isogai. On the transonic-dip mechanism of flutter of a sweptback wing. *AIAA Journal*, 17(7):793–795, 1979.

[22] K. Isogai. Transonic-dip mechanism of flutter of a sweptback wing: Part II. *AIAA Journal*, 19(9):1240–1242, 1981.

[23] R.L. Bisplinghoff, H. Ashley, and R. L. Halfman. *Aeroelasticity*. Dover Publications, 1996.

[24] F. Liu, J. Cai, and Y. Zhu. Calculation of wing flutter by a coupled fluid-structure method. *Journal of Aircraft*, 38(2):334–342, 2001.

[25] J.J. Alonso and A. Jameson. Fully-implicit time marching aeroelastic solution. In *AIAA 32nd Aerospace Sciences Meeting and Exhibit*, 10-13 January, 1994.

[26] Z. Biao, Q. Zhide, and G. Chao. Transonic flutter analysis of an airfoil with approximate boundary method. In *26th international congress of the aeronautical sciences*, 2008.

[27] J.T. Thomas, K.C. Hall, and E.H. Dowell. Reduced-order aeroelastic modeling using proper-orthogonal decomposition. *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, 1999.

[28] C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H. Matthies. Nonlinear fluid-structure interaction problem. Part I : implicit partitioned algorithm, nonlinear stability proof and validation examples. *Computational Mechanics*, 47:305–323, 2011.

[29] M. Olivier, G. Dumas, and J. Morissette. A fluid-structure interaction solver for nano-air-vehicle flapping wings. In *Proceedings of the 19th AIAA computational fluid dynamics conference*, pages 1–15, San Antonio, USA, June 2009.

[30] E.C. Yates. Agard standard aeroelastic configuration for dynamic response I - Wing 445.6. *AGARD Report 765*, 1988.

[31] G.S.L. Goura. *Time marching analysis of flutter using computational fluid dynamics*. PhD thesis, University of Glasgow, 2001.

[32] R. Beaubien, F. Nitzsche, and D. Feszty. Time and frequency domain solutions for the AGARD 445 wing. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Munich, Germany, 2005.

[33] B. Zhang, W Ding, J. Shengcheng, and J. Zhang. Transonic flutter analysis of an AGARD 445.6 wing in the frequency domain using the Euler method. *Engineering applications of computational fluid mechanics*, 10(1):244–255, 2016.

[34] R.B. Melville, S.A. Morton, and D.P. Rizzetta. Implementation of a fully-implicit, aeroelastic Navier-Stokes solver. In *13th Computational Fluid Dynamics Conference*, Snowmass Village, CO, USA, June-July 1997.